

Date=3/12/2020

Lecture By=Manish Mahant

Subject ⇒ Project Day-2

IN PREVIOUS LECTURE (QUICK RECAP) Date-2/12/2020	In Today's Lecture (Overview)
Google APIs About Local Storage Syntax Example Browser compatibility	Login With Google Using ReactJS Create ReactJS Project Add Routing in ReactJS Add Google Sign-In and Sign-Out Button and Get the AccessToken

Login With Google Using ReactJS

Create ReactJS Project

Create a ReactJS project by using the following command:

Shell

```
npx create-react-app sociallogin
```

1

Open the newly-created project in Visual Studio Code and install Reactstrap and Bootstrap in this project by using the following commands respectively. Learn more about [Reactstrap](#).

Shell

```
npm install --save bootstrap
```

1

```
npm install --save reactstrap react react-dom
```

2

Now, open the index.js file and add import Bootstrap.

Shell

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

1

Now install the [Axios library](#) by using the following command. Learn more about [Axios](#).

Shell

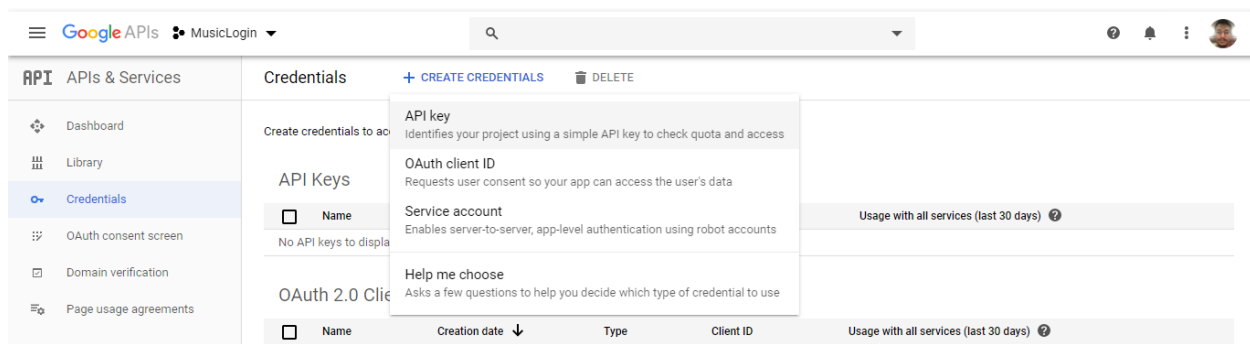
```
npm install --save axios
```

1

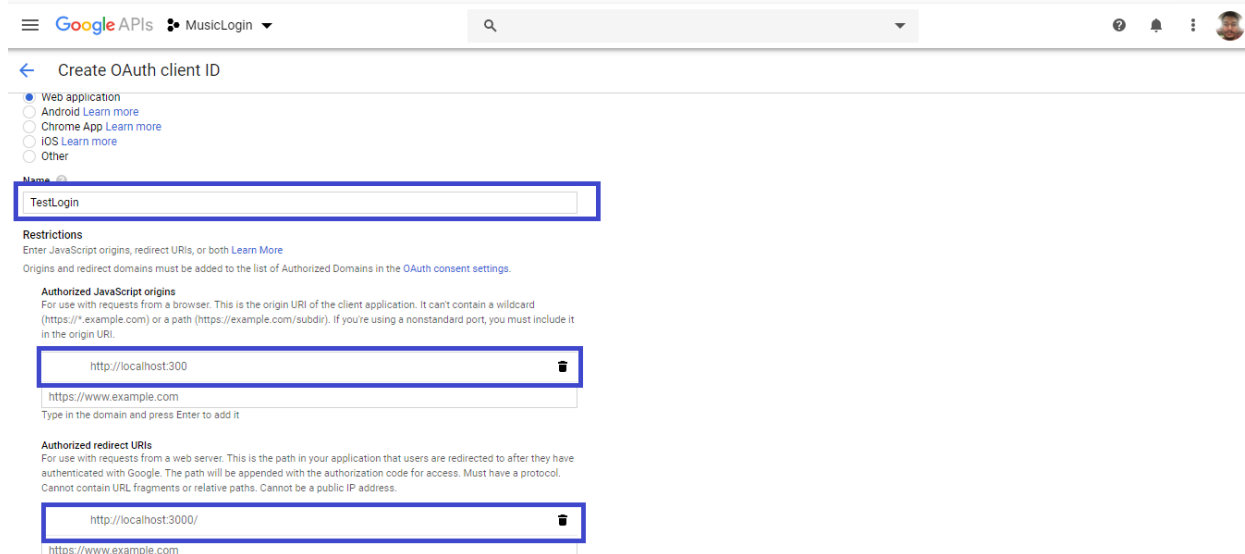
Now install the `react-google-login` React plugin using the following command:

```
npm install react-google-login --save
```

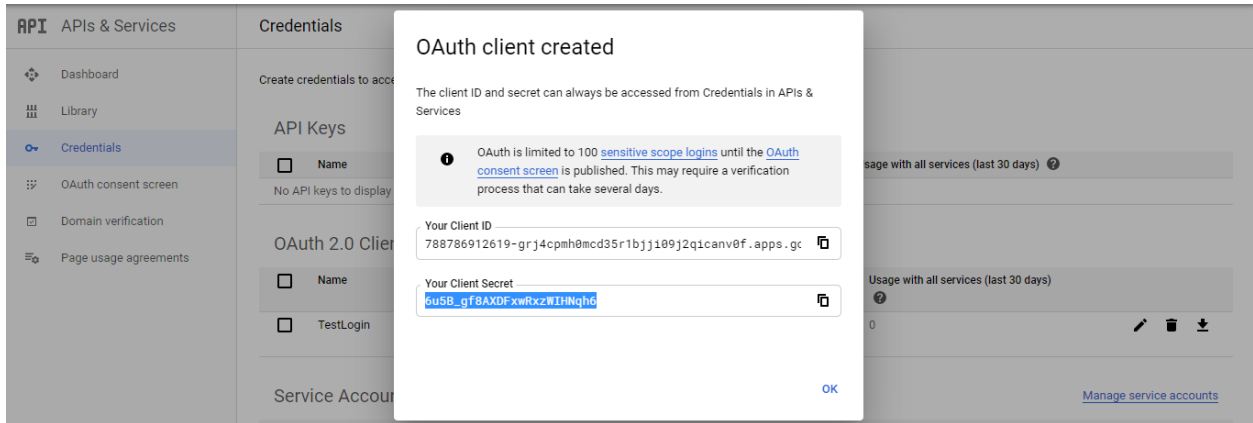
Create a Google App and select "Get Client ID." The first thing we need is to create a Google Project to get user credentials. Go to the [Google API Console](#) and click on **Credentials**. Click on **Create Credentials** and choose "OAuth client ID."



Select **web application**, enter your project URL, and click on the **Create** button.



It will create a client ID and secret key.



Now, in Visual Studio Code, go to the src folder and create a new folder. Inside this folder add two new components: Loginbygoogle.js, and Dashboard.js.

Add Routing in ReactJS

Install react-router-dom package by using the following command:

Shell

```
npm install react-router-dom --save
```

Open app.js file and imports of Router and Route (react-router-dom) and two components:

JavaScript

```
import Loginbygoogle from './Loginbygoogle'

import Dashboard from './Dashboard';

import { BrowserRouter as Router, Switch, Route, Link } from
'react-router-dom';
```

Add the following code in app.js file:

JavaScript

```
import React from 'react';
import logo from './logo.svg';
import './App.css';
import Loginbygoogle from './Loginbygoogle'
```

```
import Dashboard from "../Dashboard";
import { BrowserRouter as Router, Switch, Route, Link } from
'react-router-dom';
function App() {
  return (
    <>

    <div className="App">

      <Router>

        <div className="container">

          <Switch>

            <Route exact path="/" component={Loginbygoogle} ></Route>

            <Route path="/Dashboard" component={Dashboard} ></Route>

          </Switch>

        </div>

      </Router>

    </div>

    </>

  );
}
export default App;
```

Now, open the Loginbygoogle.js file and add the following code.

```

1 import React, { Component } from 'react'
2 import FacebookLogin from 'react-facebook-login';
3 import GoogleLogin from 'react-google-login';
4 import { Redirect } from 'react-router-dom';
5 import axios from 'axios'
6 export class Loginbygoogle extends Component {
7   constructor(props) {
8     super(props);
9     this.state = {
10     };
11     // this.signup = this
12     //   .signup
13     //   .bind(this);
14   }
15   signup(res) {
16     const googleresponse = {
17       Name: res.profileObj.name,
18       email: res.profileObj.email,
19       token: res.googleId,
20       Image: res.profileObj.imageUrl,
21       ProviderId: 'Google'
22     };
23     debugger;
24     axios.post('http://localhost:60200/Api/Login/SocialmediaData', googleresponse)
25       .then((result) => {
26         let responseJson = result;
27
28         sessionStorage.setItem("userData", JSON.stringify(result));
29         this.props.history.push('/Dashboard')
30       });
31   };
32   render() {
33     const responseGoogle = (response) => {
34       console.log(response);
35       var res = response.profileObj;
36       console.log(res);
37       debugger;
38       this.signup(response);
39     }
40     return (
41       <div className="App">
42         <div className="row">
43           <div className="col-sm-12 btn btn-info">
44             Login With Google Using ReactJS
45           </div>
46         </div>
47         <div className="row">
48           <div style={{ 'paddingTop': '20px' }} className="col-sm-12">
49             <div className="col-sm-4"></div>
50             <div className="col-sm-4">
51               <GoogleLogin
52                 clientId="788786912619-k4tb19vgofvmn97q1vsti1u8fnf8j6pa.apps.googleusercontent.com"
53                 buttonText="Login with Google"
54                 onSuccess={responseGoogle}>

```

For More About it click The link Below

<https://dzone.com/articles/login-with-google-using-reactjs#:~:text=Create%20a%20Google%20App%20and%20select%20%22Get%20Client%20ID.%22,choose%20%22OAuth%20client%20ID.%22&text=Select%20web%20aapplication%2C%20enter%20your,click%20on%20the%20Create%20button.>

Add Google Sign-In and Sign-Out Button and Get the AccessToken

There is a great React package `react-google-login` can help us to integrate Google Sign-In and Sign-Out button to our React App easily. Let's install it.

```
npm install react-google-login
```

`cd src` and add a new file named `GoogleBtn.js`, add the following code and your Google Client ID:

```
import React, { Component } from 'react'
import { GoogleLogin, GoogleLogout } from 'react-google-login';

const CLIENT_ID = '<your Client ID>';

class GoogleBtn extends Component {
```

```
constructor(props) {
  super(props);

  this.state = {
    isLoggedIn: false,
    accessToken: ''
  };

  this.login = this.login.bind(this);
  this.handleLoginFailure = this.handleLoginFailure.bind(this);
  this.logout = this.logout.bind(this);
  this.handleLogoutFailure = this.handleLogoutFailure.bind(this);
}

login (response) {
  if(response.accessToken){
    this.setState(state => ({
      isLoggedIn: true,
      accessToken: response.accessToken
    }));
  }
}

logout (response) {
  this.setState(state => ({
    isLoggedIn: false,
    accessToken: ''
  }));
}

handleLoginFailure (response) {
  alert('Failed to log in')
}

handleLogoutFailure (response) {
  alert('Failed to log out')
}

render() {
  return (
```



```
<div>
  { this.state.isLogined ?
    <GoogleLogout
      clientId={ CLIENT_ID }
      buttonText='Logout'
      onLogoutSuccess={ this.logout }
      onFailure={ this.handleLogoutFailure }
    >
    </GoogleLogout>: <GoogleLogin
      clientId={ CLIENT_ID }
      buttonText='Login'
      onSuccess={ this.login }
      onFailure={ this.handleLoginFailure }
      cookiePolicy={ 'single_host_origin' }
      responseType='code,token'
    />
  }
  { this.state.accessToken ? <h5>Your Access Token: <br/><br/> {
this.state.accessToken }</h5> : null }

</div>
)
}
}

export default GoogleBtn;
```